RabbitMQ tutorial - "Hello World!" — Rabbit

Introduction

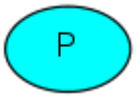
RabbitMQ is a message broker: it accepts and forwards messages. You can think about it as a post office: when you put the mail that you want posting in a post box, you can be sure that Mr. or Ms. Mailperson will eventually deliver the mail to your recipient. In this analogy, RabbitMQ is a post box, a post office and a postman.

The major difference between RabbitMQ and the post office is that it doesn't deal with paper, instead it accepts, stores and forwards binary blobs of data – *messages*.

> Prerequisites
>
> This tutorial assumes RabbitMQ is installed and running on `localhost` on standard port (`5672`). In case you use a different host, port or credentials, connections settings would require adjusting.
>
> Where to get help
>
> If you're having trouble going through this tutorial you can contact us through the mailing list.

RabbitMQ, and messaging in general, uses some jargon.

Producing means nothing more than sending. A program that sends messages is a *producer* :



A queue is the name for a post box which lives inside RabbitMQ. Although messages flow through RabbitMQ and your applications, they can only be stored inside a *queue*. A *queue* is only bound by the host's memory & disk limits, it's essentially a large message buffer. Many *producers* can send messages that go to one queue, and many *consumers* can try to receive data from one *queue*. This is how we represent a queue:

queue_name